TITLE OF INVENTION

Inventor: Shaik Cheman
Citizenship: Indian
City of Residence: Riyadh, Saudi Arabia.

Absolute public key cryptographic system and method surviving private-key compromise with other advantages

CROSS-REFERENCE TO RELATED APPLICATIONS

Not Applicable

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

Not Applicable

REFERENCE TO A MICROFICHE APPENDIX

Not Applicable

BACKGROUND

Eavesdropping is intercepting the data traversing the Internet at an intermediary point and reading the contents. Eavesdropping is becoming possible on the Internet because the data from one end has to travel to the other through a number of intermediary nodes called routers, which are neither under the control of the sender nor under that of the recipient at the destination. Eavesdroppers use a sniffer to intercept the data arriving at a router en route the destination. A sniffer is a program and/or device that monitor data passing through a network. Sniffers are easily available in the market as tools providing legitimate network management functions. Unfortunately hackers use them as their favorite honed weapons for stealing information off a network. Unauthorized sniffers can be extremely dangerous to a network's security, as they are virtually impossible to detect.

Cryptography is a wise technique widely employed in protecting Internet communications and e-commerce transactions to defeat eavesdropping. Basically it involves two steps - data encryption on the sender side and decryption on the recipient side. Cryptography algorithms are classified into Symmetric and Asymmetric, or Private-key and Public-key. Private–key algorithms use the same key for both encryption and decryption, and are not suitable for today's web–based systems involving many strange participants everyday. It is hard to share secret keys and they need a secure channel for distribution.

The shortcomings of the private-key algorithms are overcome by the public-key algorithms, which use different keys, called public and private, for encryption and

decryption. The two keys are mathematically related to each other, and not easily deducible one from the other.

One of the best known and most widely used public key algorithms is the **RSA algorithm** named for its creators Rivest, Shamir, and Adleman. The original RSA algorithm is described in U.S.Patent No. 4,405,829, entitled "Cryptographic Communications System and Method" issued on Sep.20, 1983 in the names of Rivest, Shamir, and Adleman. This patent is incorporated by reference as background information.

The RSA algorithm for encryption and decryption is given as follows:
RSA Encryption: C = M.sup.e mod n, where M is original message and C is ciphertext.
RSA Decryption: M = C.sup.d mod n, where p and q are two prime numbers and n = p.q, and e is a number relatively prime to (p-1).(q-1).

The value (p-1).(q-1) is called Euler Totient Function of n and represented by phi.
mod operator represents the remainder left when the left hand operand is divided by the right hand operator.

d is called multiplicative inverse of e, which satisfies the relation e.d = k.phi +1 for any suitable k. For large e values d value can be computed using Extended Euclid's Algorithm. p, q and phi are discarded once d value is computed. The numbers (e,n), called public key, is revealed to the public. d is called private key and maintained in strict confidence. To compute d from (e,n), one has to perform nearly $\sqrt{n}$ division operations, which would take several years for large keys.

Data encrypted by one can be decrypted only by the other. Encryption and decryption involve exponential modular arithmetic operations on a number that is a function of the original message.

Public-key cryptography has emerged into a superior technology over the Private-key cryptography because of its suitability to e-commerce with its capabilities like data integrity, authentication and non-repudiation. Another public key algorithm widely known is **ECC**(Elliptic Curve Cryptography).

Unfortunately, even the public key cryptography has its own shortcomings. A weakness of the present-day public key algorithms is that they do not survive the private-key compromise attacks following an internal breach of trust. In reality, this is what is happening in today's competitive business environment. Security administrators of well established e-commerce companies resorting to accept the lucrative bribes offered by the competitors make void the security potential of the present day public-key cryptography algorithms like RSA and ECC. Once a breach takes place in business and subsequently the private key of the business is revealed, the public-key algorithms become absolutely useless, because the degree of security that RSA and ECC offer to communications after private key compromise is nothing more than zero. The revealed key will be used by the

competitor, or the eavesdropper to decipher the intercepted data at an intermediary router. If a Certifying Authority's private key itself is compromised, the event should be considered catastrophic. Immediately, the CA must cease issuing new certificates under the key and the old certificates must be recalled and reissued using a new key.

Another weakness of public key algorithms is they secure only the public-to-private-side communications and fail to protect the private-to-public-side communications. To illustrate, suppose Bob, Chris and David are sharing Alice's public key. When Bob sends a message to Alice, Chris and David can not eavesdrop on their communication, as they do not know Alice's private key, which is necessary to decrypt the data. But the converse is not true, that is, when Alice sends a message to Bob, Chris and David can eavesdrop on the communication and successfully read the message. This is because Chris and David share the same public key of Alice with Bob, which is necessary for decryption this time.

The mathematical approach of breaking RSA cipher text is to factor the key modulus, a very large number, into two prime numbers. This requires several years of computation, some times even millions of years. For large keys, this is too difficult a task for the hackers and is quite impractical unless the hackers group has several thousands of machines engaged in parallel computing. However, there is another human approach to breaking RSA, ECC or any other such cipher text - pulling into breach of trust the security administrators or the private key guardians of a business organization, who most times, if not always, yield to material and financial gains. This weakness of the public key algorithms calls for a better concept and approach towards performing the cryptographic operations on Internet communications.

BRIEF SUMMARY

This invention relates to cryptographic systems, computers, and computer-implemented methods for performing encryption and decryption operations. More particularly, this invention relates to a cryptographic system that survives private key compromise and provides two-way communication security and also greater security. This system also allows smaller keys to be used for mobile devices with less powerful processors.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG.1 Cryptographic process of the method
FIG.2 Routing of different encrypted versions of a message on Internet
FIG.3 Flow chart showing encryption and decryption steps of algorithm.1
FIG.4 Flow chart showing encryption and decryption steps of algorithm.2
FIG.5 Flow chart showing steps of key computation for algorithm.1
FIG.6 Flow chart showing steps of key computation for algorithm.2

DETAILED DESCRIPTION OF THE INVENTION

Absolute Public-key Cryptography is a new technique, which overcomes the shortcomings of the public key cryptography and survives private key compromise. It

relieves the private key owner from the burden of maintaining strict confidentiality of the key and protects the business from undergoing bankrupt after an internal breach of trust. While surviving private key compromise, it overlays an extra layer of security on electronic and mobile commerce transactions and Internet communications when the private key is maintained in confidence.

Another advantage of Absolute public-key cryptography is it provides two-way communication security. When Alice sends message encrypted by her private key to Bob, only Bob can decrypt it with Alice's public key. Chris and David, who share Alice's public key with Bob can not decrypt the message. Public key cryptography algorithms do not support this private-to-public side communication security.

In Absolute Public-key Cryptography, both public key and private key contain two or more components. Data are encrypted into two or more versions depending upon the number of key components, and each version is delivered to the destination as a separate set of packets with some time gap. All these packet sets are received at the destination host and passed through a mathematical process that decrypts them into the original message. Over each of these versions an exponential modular operation is performed with the corresponding private key component, and finally a multiplicative modular operation is performed among all these resulting values to extract the original message. All the versions are identified as belonging to the same message by means of a message identifier, which is the same in all message versions. The message identifier is not encrypted by the sender. All the encrypted versions of data are necessary to obtain the original message. Even a single version missing produces junk data in the decryption process. This serves as a security advantage for the data transfer between the two ends. Private key in Absolute Public-key Cryptography is called Robust Private-key since it is invulnerable to compromise attacks.

Absolute Public Key Cryptography enjoys the Internet routing architecture. It derives its extra strength and security potential from the routing mechanism of packet switching net works on the Internet. To better appreciate the algorithm, it is necessary to understand the traffic routing model of the Internet and the packet switching protocols for data delivery on the Internet.

The Internet is constituted of several packet switching networks. Packet switching refers to protocols in which messages are fragmented into small packets. Each packet is transmitted individually across the net. Each packet traverses a number of networks along the way. If several messages are sent, all to the same recipient, they may each travel a different group of networks. In fact, all the packets within a single message may not travel the same route of networks. This is what makes the Absolute Public-key Algorithm extensively strong against private-key compromise due to internal breach of trust or lack of proper security measures in securing the private key.

Each packet delivered from the sender follows a convenient route to its destination. Along this convenient route the packet has to pass through a number of intermediary routers. The next hop from the current router in the convenient route is decided by the

information in routing tables of the current router. The routing information changes every nanosecond at a router depending upon the load and traffic conditions at the surrounding routers, thereby driving different packets through different routes to their final destination.

An eavesdropper intercepting the traffic at a router does not have all the data versions available at the same router. Even in a situation where a single version of the message is missing, an eavesdropper's decryption at the router produces a sheer junk. Decryption will not be complete until all versions of data participate in the process.

In the implementing algorithms, encryption can be performed in two ways while decryption process is the same in both. The two types of encryption are Blind-key Encryption and Relative Composite key Encryption.

Blind-key Encryption and Decryption Algorithm (Algorithm.1)

In this scheme, the relations between the keys components are as follows.

e.sub.1 .d.sub.1 +e.sub.2 .d.sub.2 = k.sub.1 .phi +1 Eqn(1)

d.sub.1 + d.sub.2 = k.sub.2 .phi, Eqn(2)

where

phi - Euler Totient Function of n, the key modulus
e.sub.1, e.sub.2, n - Public key
d.sub.1, d.sub.2, n – Robust Private key
k.sub.1, k.sub.2 – Suitable integers

phi and n are inter related as follows
n = p.q, where p and q are two prime numbers
phi = (p-1).(q-1)
p, q and phi are discarded after keys are computed.

Actually when a message is encrypted, it is not encrypted directly with the original public key (e.sub.1, e.sub.2, n). Instead, it is encrypted with a blind key (e.sub.1 +t, e.sub.2 +t, n) where t is a random number generated on the sender's machine before encryption. t is discarded after encryption is completed and not passed to the receiver along with the message.

Encryption and decryption are performed according to the following steps.

The sender encrypts his message M into two versions M.sub.1 and M.sub.2 by performing exponential modular operations on M, where

M.sub.1 = M.sup.(e.sub.1 +t) mod n and Eqn(3)
M.sub.2 = M.sup.(e.sub.2 +t) mod n Eqn(4)

M.sub.1 and M.sub.2 are delivered to the receiver

The receiver computes N.sub.1 and N.sub.2 similarly, where

N.sub.1 = (M.sub.1).sup.(d.sub.1) mod n and Eqn(5)

N.sub.2 = (M.sub.2).sup.(d.sub.2) mod n Eqn(6)

Next, the receiver performs multiplicative modular operations on N.sub.1 and N.sub.2 and computes N, where

N = N.sub.1 .N.sub.2 mod n Eqn(7)

This ends the cryptographic process on both sides. At the end of the above process the N value obtained by the receiver is the same as M, which is the original message.

The equality N = M may be proved **mathematically** as follows.

From Eqn(7) N = N.sub.1. N.sub.2 mod n
Substituting for N.sub.1 and N.sub.2 in the above from Eqn(5) and Eqn(6),

N = [(M.sub.1).sup.(d.sub.1) mod n . (M.sub.2).sup.(d.sub.2) mod n] mod n

According to modular arithmetic properties the above equation simplifies to

N = [(M.sub.1).sup.(d.sub.1) .(M.sub.2).sup.(d.sub.2)] mod n

Substituting for M.sub.1and M.sub.2 in the above from Eqn(3) and Eqn(4),

N = [(M.sup.(e.sub.1 +t) mod n).sup.(d.sub.1) .
(M.sup.(e.sub.2 +t) mod n).sup.(d.sub.2)] mod n

Again, according to modular arithmetic properties the above equation simplifies to

N = [(M.sup.(e.sub.1 +t)).sup.(d.sub.1) .
(M.sup.(e.sub.2 +t)).sup.(d.sub.2)] mod n

Rearranging the exponents in the above,

N = [(M.sup.(e.sub.1 . d.sub.1)) .(M.sup.(e.sub.2 . d.sub.2)) .
(M.sup.(t. d.sub.1 + t. d.sub.2))] mod n

Substituting Eqn(1) and Eqn(2) in the above,

N = [(M.sup.(k.sub.1 .phi +1 + t. k.sub.2. phi)] mod n

Simplifying the exponent in the above,

N = [(M.sup.(k.sub.3 .phi +1)] mod n, where k.sub.3 = k.sub.1 + t. k.sub.2

According to Euler's theorem in Number theory,

M.sup.(k.phi +1) mod n = M for any M, k and n when M and n are relatively prime, where phi is the Euler Totient Function of n.

Hence, N = M

Interestingly, it can be seen from the above proof that the effect of blinding the original public key with a random number t on the sender's machine is neutralized when both versions of the message participate in the process of decryption. This feature of the algorithm imparts an excellent security to the messages, because an eavesdropper at an intermediary router with access only to one version of the message can not decrypt it unless he gets hold of the other version. Locating the other version is very difficult as it might have followed any other route on the Internet. As we have learned previously in this document, a packet route is decided based upon the traffic condition at different routers around at the time of the packet arrival.

It is suggested that, to achieve better security, the two versions of the message must be delivered with a time gap of few nanoseconds. An eavesdropper who wants to decrypt the message with a single encrypted version of it has to try different possible values of t, the blinding number. A multiplicative inverse has to be computed for each e.sub.1+t and e.sub.2 +t and an exponential modular operation needs to be performed with each one of these as exponents. One of these exponents will yield the original message. But the eavesdropper never knows which particular t value yields the right message. The searching range of t for an eavesdropper is (-min(e.sub.1, e.sub.2) to phi). Since the order of phi is same as n, an eavesdropper has to perform nearly n mathematical operations where as RSA system requires only $\sqrt{n}$ operations. Therefore the security factor of this cryptographic system is $\sqrt{n}$. For example if a 512 bit key is used for encryption, breaking the cipher of this system takes more than 2.sup.256 times the period as required by RSA system.

RSA algorithm requires only factoring of n into p and q for finding the private key. Once factoring becomes successful, phi is computed, and then the private key is computed as multiplicative inverse of the public key with respect to phi. This requires only $\sqrt{n}$ division operations. Here, for attacks on Absolute public key cryptography messages, an eavesdropper with a single message version needs to run the Extended Euclid's algorithm phi times for finding the multiplicative inverses of all the possible e.sub.1+t and e.sub.2+t values. Each run of the algorithm involves several subtraction, multiplication and division operations. Once the multiplicative inverses are found, the attacker has to try decryption with each of these as exponents. Each decryption involves several exponential modular operations, which are very expensive than the division operations as required in

the RSA attacks. More over there is no way of identifying the original message out of all these resulting decryptions. Even the compromised robust private key will not help in decrypting the message until the attacker acquires both versions of the message.

The following simple illustration demonstrates the Public-key Cryptography Algorithm.

Let p = 3 q = 5 n = p.q = 3.5 = 15 phi = (3-1).(5-1) = 8

The values 4, 5, 7 and 11 for e.sub.1, d.sub.1, e.sub.2 and d.sub.2 respectively satisfy the two necessary key component equations -

e.sub.1 .d.sub.1 + e.sub.2 .d.sub.2 = k.sub.1 .phi + 1

d.sub.1 + d.sub.2 = k.sub.2 .phi

where k.sub.1 = 12 and k.sub.2 = 2.

Let the original message be a single character 'C' for simplicity.
Map the character 'C' to the number 3. So M = 3
Select a random number t = 3

Now the Blind encryption key (e.sub.1 +t, e.sub.2 +t, n) ≡ (7, 10, 15)
M.sub.1 = 3.sup.7 mod 15 = 12
M.sub.2 = 3.sup.10 mod 15 = 9

When the receiver receives the values 12 and 9, he will perform the following computations using the Robust Private-key (5, 11, 15)

N.sub.1 = 12.sup.5 mod 15 = 12

N.sub.2 = 9.sup.11 mod 15 = 9

M = N = 12.9 mod 15 = 108 mod 15 = 3

Hence, the original message value 3 has been extracted which may be mapped again to 'C'.

Relative Composite-key Encryption and Decryption Algorithm (Algorithm.2)

In Relative Composite-key Algorithm, no blinding is done with a random number. Instead, the key components e.sub.1 and e e.sub.2 are so selected that each of them has a separate common factor with phi, but relatively prime to each other. Note that if both common factors are not relatively prime to each other, a private key does not exist for decryption by the intended recipient.

The key equation to be satisfied in this scheme is simply

e.sub.1 .d.sub.1 + e.sub.2 .d.sub.2 = k.sub.1 .phi + 1

where again

phi- Euler Totient Function of *n*, which is the key modulus
e.sub.1, e.sub.2, n - Public key
d.sub.1, d.sub.2, n – Robust Private key
k.sub.1 – A suitable integer

As an example the following set of values satisfy the above equation

p = 3 q = 7 n = 21 phi = 12 k.sub.1 = 5

e.sub.1 = 3 d.sub.1 = 17 e.sub.2 = 2 d.sub.2 = 5

Here, encryption is performed without blinding the public key and two versions of data are generated. Decryption is performed by the recipient following the same steps as in the previous algorithm. Still it guarantees good security against attacks on a single version of the message. While it can be used for e-commerce transactions where large size keys are affordable, it is especially suitable for mobile commerce transactions, because this allows small keys for encryption and decryption. Though the keys are small, there is no threat from computational aspect. In fact, when keys are small finding a multiplicative inverse for e.sub.1 and e.sub.2 individually and trying to decrypt a single version of the message into the original one requires only a few milliseconds as in the RSA algorithm. But fortunately e.sub.1 and e.sub.2 as selected in this encryption scheme do not have individual multiplicative inverses as they have a common factor with the Euler Totient Function phi. Even the compromised robust private key will not help the eavesdropper until he obtains all the encrypted message versions. With a single version of the message missing, an attacker will not be able to break the cipher text.

The small length keys of this scheme are especially useful for encryption in mobile devices like GSM hand sets and smart cards as they lack the ability to perform heavy exponential modular arithmetic operations with large keys.

The following simple illustration demonstrates the cryptographic process involved.

Let the original message be a single character 'C' for simplicity.
Map the character 'C' to the number 3. So M = 3

Now the encryption key (e.sub.1, e.sub.2, n) ≡ (3, 2, 21)
M.sub.1 = 3.sup.3 mod 21 = 6
M.sub.2 = 3.sup.2 mod 21 = 9

When the receiver receives the values 12 and 9, he will perform the following computations using the Robust Private-key (17, 5, 21)

N.sub.1 = 6 sup.17 mod 21 = 6

N.sub.2 = 9.sup.5 mod 21 = 18

M = N = 6.18 mod 21 = 108 mod 21 = 3

Hence, the original message value 3 has been extracted which may be mapped again to 'C'.

Multiple Component Keys

As we have discussed earlier in this document, the number of components of the keys need not be limited to only two. Multiple component keys can be used to further improve the security and speed. The governing key equations in this case for blind key encryption and decryption are as follows:

SUM(e.sub.i .d.sub.i) = k.sub.1 .phi+ 1 and

SUM(d.sub.i) = k.sub.2 .phi, where i = 1 to r.

The first equation is called orthogonality of encrypting and decrypting keys while the second may be referred as condition of equilibrium among the decrypting key components.

The governing key equation for the Relative Composite-key Encryption and Decryption scheme is as follows:

SUM(e.sub.i .d.sub.i) = k.sub.1 .phi+ 1

Every e.sub.i must be selected such that it has a common factor with phi. But there should not be a common factor among all e.sub.i and phi. This condition may be referred as relative composite ness of encrypting key components.

Multiple component keys further impart a significant improvement in security and speed for e-commerce as well as m-commerce transactions. When a small size multiple component key is used, it enormously brings down the computational overhead while increasing the size of data produced on encryption. Large size data delivered in wireless communications may not cause a striking affect and can be overweighed by the speed and security achieved in mobile communications. Even in e-commerce communications this may not cause much traffic overhead on the IP routers as there are several thousands of IP routers on the Internet.

Private-to public-side communication security

Communication from private to the public side can be secured by using a separate key pair for encryption and decryption. While computing this key pair, the equilibrium

condition should be imposed on the public key, which is the decrypting key this time. This allows blinding of the private key components and thereby prevents one eavesdropping on other's message among the public. In case of Relative Composite key Algorithm, relative compositeness should be imposed on the private key components.

Key Generation

Large size encryption and decryption keys for both the algorithms can be generated using Extended Euclid's Algorithm, which involves computing greatest common divisors and multiplicative inverses. The key generation process for both the algorithms is shown in FIG 5.and FIG.6. and explained in detail in detailed description of the drawings.

Redundant Data Delivery – A Security Improvement Concept

Redundant Data Delivery tremendously improves security of electronic and mobile commerce transactions. In this, redundant versions of data are generated and delivered to the recipient. Before performing encryption the sender receives a number of version identifiers equal to the number of components in the public key and a single message identifier. The sender generates more encrypted data versions than the number of components in the key. For excess data versions, no computations are performed. They are generated just by random filling of bytes. The genuine versions of data are tagged with the version identifiers sent by the receiver while the redundant data versions are tagged with some fooling version identifiers by the sender. When the receiver receives all the data versions, both genuine and redundant, he selects only the genuine versions according to his record of version identifiers against the message identifier and decrypts them into the original message. An eavesdropper who has no previous record of these message and genuine version identifiers can not distinguish between the genuine and redundant versions.

Even for an aggressive eavesdropper or a hacker who has a mechanism by which he can get hold of all the data packets for a given destination, it would be very difficult to break the cipher text. In Relative Composite-key Encryption scheme even a key as small as 10 bits long, with 15 components e.sub.1, e.sub.2, ...e.sub.15, and 15 redundant generated versions gives excellent security and speed to the mobile communications and smart card transactions. An eavesdropper in this case has to select 15 genuine versions out of total 30 versions leading to **30-C-15** or nearly 1.5 billion combinations and needs to perform the same number of multiplicative modular operations to extract the original message. Elimination of the source address in the packet headers further shoots up the combinations to be selected by the eavesdropper. Because, now the amount of traffic at the receiver's host from which the combinations have to be selected is pretty high as other packets get mingled with the actual message packets. Source address may be included in the message itself for identifying the sender.

Anonymity as Another Means of Security

The security of the communications may be further improved by collecting the encrypted data versions at two different hosts. In this method at least one version is received at a secret host, which is not transparent to the public and all the remaining versions are received at the main host, which is usually the business server. The encrypted versions received at the secret host may be further encrypted by a symmetric key encryption algorithm like DES and reflected to the main host, where it would be decrypted by the same algorithm before participating in the process of Absolute public key algorithm. Symmetric key encryption by DES would not cause any practical problems like key management, as the same party owns both the hosts. As at least one version is shrouded by DES encryption, even an eavesdropper capturing all the versions at the main receiving host can not extract the original message.

For maintaining a secret host, one may have a dynamic IP address allocated for his machine by his ISP. Another means to hide the secret server from the public is to use Network Address Translation, which allows an organization to present itself to the Internet with one address. As a result people can not identify computers on network and capture any details of their location, IP address etc.

Implementing at Browser Level

Absolute Public-key Algorithm can be implemented within browsers like Netscape Navigator and Internet Explorer to perform encryption in the background and avoid any user interaction.

Digital Signatures and Certificates

Similar to public key cryptography absolute public key algorithm also supports digital signatures for non-repudiation and digital certificates for authentication. Hash functions can be run on the original messages to generate a message digest to verify the originality of the messages received.

FIG.1 shows the cryptographic process of the method. It shows the encrypting and decrypting keys with their components. Each component of the encrypting key works on the message and generates an individual encrypted version of it. Step.10 shows the encryption and the output of it as different message versions M.sub.1, M.sub.2,..., M.sub.r. Step.20 shows passing of these versions independently to the Internet with a small time gap between every two consecutive versions. Each of these versions travel independently across the Internet medium and finally reach the destination machine. Step.30 shows the collection of these versions at the destination. In step.40, over each of these versions an exponential modular operation is performed with the corresponding decrypting key components. The output of stpe.40 is shown as N.sub.1, N.sub.2,..., N.sub.r. Step.50 shows the final decryption of these values into the original message, which involves a multiplicative modular operation on these values.

FIG.2 shows the routing of different encrypted versions of a message on the Internet. It shows the network of several IP router on the Internet, through which different encrypted versions of the message take different routes to the destination depending upon the condition and traffic load at different routers on the Internet. The three types of arrows represent three different encrypted versions of the message. Each sequence of arrows shows the route of an encrypted message version. Point.1 represents the sender's machine. Point.2 represents the sender's router through which he directly connects to the Internet. Point.3 represents an intermediary router on the Internet, which is not under the sender's or recipient's control. Point.4 represents the recipient's router while point.5 represents the recipient's machine, where all the encrypted versions are collected and decrypted in to the original message.

FIG.3 shows the flow chart of algorithm.1 of the method in which blinding is performed on the encrypting key components before encryption. Shown at the top are the encrypting and decrypting keys. The flow starts from step.10, where the original message is made ready for encryption. At step.20, a random number t is selected. At step.30, blinding is performed by adding the random number to each of the encrypting key components. The blinded key components are e.sub.1 +t, e.sub.2 +t,..., e.sub.r +t. Step.40 shows the exponential modular operation performed on the original message using each blinded component of the encrypting key to produce different encrypted versions of the message. At step.50, the random value t is discarded. Step.60 shows the collection of all the encrypted versions of the message at the recipient's machine. Step.70 shows the exponential modular operations performed on each of these collected versions using the corresponding decrypting key components. The output is shown as N.sub.1, N.sub.2,..., N.sub.r. Step.80 shows computing the original message from the output of step.70. The computation involves multiplicative modular operations on the output values of step.70.

FIG.4 shows the flow chart for algorithm 2 of the method in which no blinding is performed. At step.10, the original message is formed ready for encryption. At step.20, the message is encrypted into a number of versions by performing exponential modular operations with each component of the encrypting key. At step.30, all these encrypted versions are collected at the recipient's machine via the Internet. Step.40 shows the exponential modular operations on each of these versions using the corresponding decrypting key components. The output is shown as N.sub.1, N.sub.2,..., N.sub.r. Step.50 shows computing the original message from the output of step.40. The computation involves multiplicative modular operations on the output values of step.40.

FIG.5 shows flow chart of the steps involved in key computation for algorithm.1 of the method. At step.10 two prime numbers p and q, and an integer k.sub.2 are selected. At step.20, the key modulus n and the Euler Totient Function phi are computed. At step.30, p and q are discarded. At step.40, d.sub.1, the first component of the decrypting key is selected. At step.50, the greatest common divisor of d.sub.1 and phi is computed. If the value is equal to 1, the control goes to step.60, else the steps.40 and 50 are repeated. At step.60, all the remaining, except the last, components of the decrypting key are selected. At step.70 the last component of the decrypting key is computed. At step.80, all the encrypting key components, except the first, are selected. At step.90, a value F is

computed, which is a function of all components of the encrypting and decrypting keys, except the first ones. Step.100 shows the computation of the first component of encrypting key, which involves finding a multiplicative inverse using Extended Euclid's Algorithm. At step.110, the Euler Totient Function, phi is discarded. At step.120, all the key components are properly arranged to form the encrypting and decryption keys.

FIG.6 shows flow chart of the steps involved in key computation for algorithm.2 of the method. At step.10 two prime numbers p and q are selected. At step.20, the key modulus n and the Euler Totient Function phi are computed. . At step.30, p and q are discarded. At step.40, all the components of the encrypting key are selected. At step.50, the greatest common divisor (gcd) of each component with the Euler Totient Function, phi is computed. If any of the gcd's is equal to 1, step.40 and 50 are repeated for that particular component. When all the gcd's computed are greater than 1, control goes to step.60, where a global gcd, G of all the previously computed gcd's is computed. If the global gcd value equals 1, control goes to step.70, else steps.40 and 50 are repeated only for the first component of the encrypting key and G is computed again. At step.70, all the decrypting key components, except the first, are selected. At step.80, a value F is computed, which is a function of all components of the encrypting and decrypting keys, except the first ones. Step.90 shows the computation of the first component of encrypting key, which involves finding a multiplicative inverse using Extended Euclid's Algorithm. At step.100, the Euler Totient Function, phi is discarded. At step.110, all the key components are properly arranged to form the encrypting and decryption keys.

REFERENCES – U.S.PATENTS

6,081,598	Cryptographic system and method with fast decryption	Dai
4,405,829	Cryptographic communications system and method	Rivest, et al

REFERENCES – OTHERS

"Cryptography And Network Security, Principles and Practice", William Stallings, Prentice Hall, Second Edition.
"Data Security", Janet Endrijonas. Prima Publishing.,1998.
"Digital Money, The new era of Internet commerce", Daniel C. Lynch, Leslie Lundquist. John Wiley & Sons, Inc., 1996.
"TCP/IP – Network Administration", Craig Hunt, O'Reilly & Associates, Inc., 1993
"Smart Cards". Henry Dreifus, J. Thomas Monk., John Wiley & Sons, Inc, 1998.